uFR Browser Extension

1.0

Digital Logic Ltd.

Address: Nemanjina 57a, 12000 Pozarevac, Serbia · Tel: +38112541022· VAT: 111385444 Reg. 21473642
e-mail: office@d-logic.com · www.d-logic.com

Table of contents

About NFC Reader Browser Extension

NFC Reader Browser Extension is a web solution aimed at simplifying usage of uFR series readers with web applications. Requirements are simple, download extension from your preferred browsers store (Chrome, Firefox, Opera) and add extension to your browser. After adding the NFC Reader Browser Extension to your browser, you will also need to register the extension in your OS by running the additional installer from the following link:
https://www.d-logic.net/code/nfc-rfid-reader-sdk/ufr-browser_extensions/tree/master/Store%20installers
In the additional settings for the extension make sure to enable "Allow access to file URLs".
You can test the NFC Browser extension by visiting our demo software page:
https://www.d-logic.net/browser-extension-demo/

Extension usage

With version 1.3.0 NFC Reader Browser Extension implements new, better and simpler approach to using uFR series readers with our browser extension.
In summary, new function **uFR_Request** has been implemented to deal with problems such as multiple asynchronous execution of commands. Proper function response is guaranted with refactored ufResponse() function.
For example, calling functions in a loop will always guarantee that untill current command receives response, next one will not be called/executed. This mechanism is implemented with a goal of keeping in sync with uFR series reader.
Since all of the function calls are to be used as asynchronous, looping will not cause browser page to be blocked while the loop is running.

Examples of how new function is used, and how to make new extension compatible with old ufRequest and ufResponse functions will be described in next paragraph, along with code snippets.

uFR_Request() examples

ufR_Request function only has one parameter and returns command response.

Prototype: await uFR_Request(**input**);

Simple example:

```
async function ReaderOpen() {
   let response = await uFR_Request("ReaderOpen");
   console.log(output);
}
```

Variable **'response'** will hold function response such as 'UFR_OK', 'UFR_READER_OPENING_ERROR' and simmilar, along with values returned, depending on a command that has been sent.
Function execution will not move on untill uFR_Request returns results.

```
async function LoopTest(e)
{
 for ( let j = 0; j < 10; j++){

   let output = await uFR_Request("ReaderUISignal 1 1");
   console.log(j);
   console.log(output);
   }
}
```

This way it will go through the loop, but, will always wait for function uFR_Request to return command response before moving on to the next function call in call stack.

Compatibility with previous versions

Compatibility of new **ufResponse** function with previous **ufRequest**(input, callback()) function can be achieved by simply making ufRequest callback function **async,** and inserting **await** keyword in front of ufResponse function
Example:

```
ufRequest("ReaderUISignal 1 1",  async function () {
   let output = await ufResponse();
   console.log(output);
  });
```

Older versions usage

For versions before v1.3.0 ufRequest and ufResponse functions are used in the following manner:

```
uFRequest(command, function () {
   let response = uFResponse();
})
```

Call UfRequest function and pass "command" parameter as string that contains UFCoder function. ufResponse function will return JSON object that contains requested data.

For further inquiries, contact us: **support@d-logic.rs**

Revision history

Date	Version	Comment
2019-04-09	1.0	Base document